Lang Learn

# Dokumentation

Das Projekt wurde in wöchentlichen Intervallen dokumentiert.

## KW 33

Da Quizlet beschlossen hat, kapitalistischen Interessen nachzugehen, habe ich mich entschlossen, den kostenpflichtigen Lernmodus in einem Konsolenprogramm wieder kostenfrei zu machen.

In dieser Woche habe ich einen einfachen "HTML-Parser" gemacht, der aus dem HTML-Code von Quizlet die Begriffe parsen kann. Zudem habe ich ein kleiner Parser geschrieben, der die Begriffe parsen kann. Das wird später nützlich, da ich vorhabe, mithilfe von "Metadaten" verschiedene Lern-Modi zu machen. Je besser mein Programm die Begriffe versteht, desto bessere Lern-Modi kann ich kreieren.

Als nächstes möchte ich Settings implementieren. Das Programm soll sehr personalisierbar sein, damit man beim Lernen am meisten profitieren kann. Dazu gehört auch später ein Serialisier / Deserialisier, den ich auch für das Speichern von Lernfortschritten benutzen möchte.

Danach muss ich noch eine geeignete "Internal-Representation" konzipieren, damit ich auf eine einfache Art und Weise auf die vorhin erwähnten Metadaten zugreifen kann.

## KW 34

In dieser Woche habe ich eine Render-API für die Windows-Konsole geschrieben. Sie kann alles, was ich benötige für diese Arbeit. Es gab Probleme mit der Abstraktion. Ich habe ein komplett neues Render-System implementiert und musste viele Fehler ausmerzen. Die API basiert auf Container. Es gibt horizontalen und vertikalen Containers, die jeweils ein Layout- oder ein Text-Container sein können. ein Layout-Container hat kann wieder weitere Layout- oder Text-Containers beinhalten. Ein Text-Container kann nur Text beinhalten.

Als zweits habe ich einen TOML-Parser ausgesucht. Ich möchte die Lern-Sets sowie die Settings im TOML-Format abspeichern. Daran werde ich noch nächste Woche arbeiten.

Weiter steht noch an, ein Lernmodus zu programmieren. Da möchte ich auch wieder mit Abstraktion arbeiten, damit in Zukunft weitere Lern-Modi hinzugefügt werden können.

## KW 35

In dieser Woche habe ich weiter am Renderer gearbeitet. Er sollte jetzt wesentlich weniger Bugs haben als zuvor. Allerdings treten immer wieder Probleme auf. Zudem ist das Rendersystem umständlich zu benutzen. Vermutlich muss ich nach einer Alternative Ausschau halten.

Des Weiteren habe ich eine einfache Version eines Lernmodus implementiert. Der Modus ist jedoch noch weit weg von Benutzerfreundlich. Daran werde ich nächste Woche noch arbeiten müssen.

## KW 36

Kleine Planänderung: Ich habe beschlossen das ganze Projekt komplett von C# auf JavaScript umzuschreiben. Das liegt daran, dass die C# Electron-Implementation schlecht ist. Ich möchte auch nicht weiter die Eigene Render-API benutzen, da immer wieder Fehler hervorkommen und die Bedienung zu umständlich ist. Zudem ist das Ganze nicht skalierbar, da meine eigene Render-API auf der Windows-Konsole basiert. Electron hingegen ist skalierbar.

In dieser Woche habe ich eigentlich mein komplettes Projekt nochmals neu geschrieben. Vom Rendering mit HTML und CSS zu dem Quizlet-Downloader bis zuletzt dem Lernmodus.

Der Parser wurde durch die Verwendung von JS deutlich einfacher, da JS einen HTML-Parser hat, den man mit Query-Befehlen zum Suchen von Übungskarten auf Quizlet missbrauchen kann.

Das Speichern von allem wird auch nicht mehr über TOML, sondern über JSON gemacht. Das liegt daran, das JSON sehr gut mit JS harmoniert.

Was am meisten Zeit in Anspruch genommen hat, war alles schön aussehen zu lassen. Dazu gibt es leider nicht viel zu sagen, ausser dass es viel Geduld braucht, ein gescheites HTML- und CSS-System zu entwickeln.

Für die Settings hat es leider nicht gereicht. Jedoch ist die aktuelle Version bereits brauchbar und hat zudem noch eine hübsche Benutzeroberfläche.

Ich habe vor die Version noch auszubessern, da ich Lang Learn dieses Schuljahr noch für das Lernen von Wörtchen benutzen will. Da ich Electron jedoch nicht so grossartig finde, werde ich vermutlich das Produkt nochmals auf ein anderes System umschreiben müssen.